(FILE 'USPAT' ENTERED AT 09:07:07 ON 23 JUN 1999)

L1	0	S	(PRE-FETCH) (W) WEG PAGE#
L2	0	S	(PRE-FETCHED) (W) WEG PAGE#
L3	0	S	(PRE-FETCHED) (2A) WEG PAGE#
L4	0	S	(PRE-FETCHED) (2A) WEB PAGE#
L5	2	S	REFRESH? (4A) WEB PAGE#
L6	2	S	REFRESH? (6A) WEB PAGE#
L7	3	S	(REFRESH? OR RELOAD?) (6A) WEB PAGE#
L8	11	S	(REFRESH? OR RELOAD?) (P) WEB PAGE#

PAT NO: 5,848,410 [IMAGE AVAILABLE] L8: 6 of 11

SUMMARY:

BSUM(25)

In one implementation, the index builder includes a file scanner to access the files; a data structure generator to generate at least one data structure and to store the index-organizing element, the descriptive element, and the location element in such data structure; and an index generator to generate the index from the stored elements. In one implementation, the index generator formats the stored elements in an HTML format to generate the index. Advantageously, this capability allows the index to be displayed to the user as a **web page** of an intranet or over the Internet. In one implementation, the file scanner periodically re-initiates processing of the files in the file system at a user-selected **refresh** rate. Advantageously, in one implementation, the file scanner initiates such periodic processing automatically without a user command.

US PAT NO: 5,790,977 [IMAGE AVAILABLE] L8: 7 of 11

DETDESC:

DETD(17)

FIG. 3 shows a **web page** 41 resulting when remote application 10 communicates with instrument 20 in a continuous **refresh** mode used for continuous data acquisition of measurement data. On a graticule 44, two overlapped traces of scaled measurement data are shown displayed. Alternatively, four or some other number of overlapped traces of scaled measurement data are displayed.

US PAT NO: 5,737,619 [IMAGE AVAILABLE] L8: 8 of 11

DETDESC:

DETD(11)

A preferred operation of the inventive method is illustrated in the flowchart of FIG. 3. The method begins at step 70 as a current **web page** is being displayed on the graphical user interface of the computer. It is assumed that this **web page** has embedded therein one or more comment tags, each of which (or perhaps several of which in combination) define an information object. Generally, although not required, each information object will be provided for one or more links in the **web page** being displayed. However, because the information object is embedded within a comment tag, it is hidden or "masked" and thus is ignored by the display routines of the browser. In step 72, the method saves or stores the information object in memory or some dedicated portion of the RAM (e.g., a cache) so that it may be easily and quickly obtained. At step 74, a test is made to determine whether a link associated with the information object has been activated. If so, the method continues at step 76 and issues a tcp/ip request to the network (assuming the link was to a URL). Step 78 represents the handshaking period during which the client waits for the appropriate response from the server. During this period, the client retrieves the information object (at step 80) and outputs the information (in step 82) to the user on the display. Steps 80 and 82 are shown in parallel to the handshaking

col. 2,

and wait step 78 to emphasize the inventive concept of displaying useful information to the viewer during the link process. At step 84 a test is then performed to determine whether the download and **refresh** of the display is complete. If so, the routine saves the information object at step 86 and opens up access to the hypertext document at step 88.

US PAT NO: 5,734,835 [IMAGE AVAILABLE] L8: 9 of 11

ABSTRACT:
A World Wide Web terminal appliance utilizes a disk drive for local storage of **Web pages** previously downloaded and rendered for display during the course of a Web surfing session. The disk drive rotates at a rate substantially in synchronization with a display **refresh** time interval of a display device, preferably a raster **refresh** cycle time of a video monitor. Therefore, the image being displayed need not occupy random-access memory, but rather is sent directly from the disk to a display interface for coupling to the display device. Little or no RAM buffering is required, so the appliance need not include a large quantity of video RAM storage. A relatively inexpensive disk is used instead, thereby bringing about advantageous cost savings.

SUMMARY:

BSUM(50)

For example, a preferred Web appliance according to the invention is used with a raster-type video display having the typical raster **refresh** period of 1/30 second, wherein every second raster line is **refreshed** during a 1/60 second period, and the interleaved raster lines are **refreshed** during the next 1/60 second. Accordingly, a 60 rotations-per-second (rps) disk allows for access of a data image (e.g., a **Web page**) stored with successive portions of the image in successive sectors of a track, in just the synchronization necessary to provide the image data from the disk as the raster is ready to receive it.

DETDESC:

DETD(12)

In accordance with the invention, the data is stored on the disk 21 in a manner which facilitates reading the data and sending it directly to the display interface. As a consequence, redisplaying a previously viewed and stored **Web page** may be accomplished without requiring a large quantity of expensive video RAM. Preferably, where the display device has a display cycle time or **refresh** time, such as that associated with raster-type video displays, the data is stored on the disk 21 so as to be accessible in sequence over a display cycle period, so that the data is read from the disk 21 just as it is needed for the display device 10.

DETDESC:

DETD(37)

Also, while the present example shows a preferred data storage arrangement for a disk rotating at the same rate as the screen **refresh**. These rates need not be the same, as long as a suitable arrangement is made for keeping track of which sector of which track which part of the **Web page** image data is kept.

US PAT NO: 5,727,129 [IMAGE AVAILABLE] L8: 10 of 11

SUMMARY:

BSUM(30)

[redacted]

Existing Web browsers maintain records of sequences of pages previously viewed during a Web surfing session. Thus, if a user wants to return to a previously-viewed page, he/she can easily do so. For instance, IBM's WebExplorer displays right and left arrows to allow a user to scroll back through previously viewed pages, and forward to the most recent page. As long as the Web terminal has the memory capacity to cache the previously viewed **Web pages**, the previously viewed pages are recovered locally. Thus, the delay of **reloading** the page over the Web is avoided.

US PAT NO: 5,572,643 [IMAGE AVAILABLE] L8: 11 of 11

DETDESC:

DETD(11)

A preferred operation of the inventive method is illustrated in the flowchart of FIG. 3. The method begins at step 70 as a current **web page** is being displayed on the graphical user interface of the computer. It is assumed that this **web page** has embedded therein one or more comment tags, each of which (or perhaps several of which in combination) define an information object. Generally, although not required, each information object will be provided for one or more links in the **web page** being displayed. However, because the information object is embedded within a comment tag, it is hidden or "masked" and thus is ignored by the display routines of the browser. In step 72, the method saves or stores the information object in memory or some dedicated portion of the RAM (e.g., a cache) so that it may be easily and quickly obtained. At step 74, a test is made to determine whether a link associated with the information object has been activated. If so, the method continues at step 76 and issues a tcp/ip request to the network (assuming the link was to a URL). Step 78 represents the handshaking period during which the client waits for the appropriate response from the server. During this period, the client retrieves the information object (at step 80) and outputs the information (in step 82) to the user on the display. Steps 80 and 82 are shown in parallel to the handshaking and wait step 78 to emphasize the inventive concept of displaying useful information to the viewer during the link process. At step 84, a test is then performed to determine whether the download and **refresh** of the display is complete. If so, the routine saves the information object at step 86 and opens up access to the hypertext document at step 88.